REMARKS

Reconsideration of the application is requested.

Applicants acknowledge the Examiner's confirmation of receipt of applicants' certified copy of the priority document being German Patent Application 198 46 676.5, filed October 9, 1998 supporting the claim for priority under 35 U.S.C. § 119.

Claims 1-5 and 8-10 remain in the application. Claims 1-5 and 8-10 are subject to examination. Claims 5, 8 and 9 have been amended. Claims 6, 7 and 11 have been canceled to facilitate prosecution of the instant application.

Under the heading "Oath/Declaration", the Examiner notes an interesting issue regarding 37 CFR 1.56. A substitute declaration will be forwarded in due time.

Under the heading "Claim Rejections - 35 USC § 102" on pages 2-6 of the above-identified Office Action, claims 1-11 have been rejected as being fully anticipated by U.S. Patent No. 5,507,028 to Liu (hereinafter Liu) under 35 U.S.C. § 102.

Liu describes a method that can be used as a branch predictor during the run of a program in a pipelined processor. A history based dynamic table is used to store information

about previously performed conditional jumps at branching points in the program. An essential feature of the method according to Liu is to set up program code sequences that were run without branching during previous applications of the program. This reduces the time which is spent to decide whether a jump to a new address must be performed or not, but does not eliminate actual decisions on the branching. Therefore, such a technique for early resolution of branch predictions depends on the registration of the branch addresses and the control of the validity of the branch conditions during the run of the program. Thus, jump addresses have to be stored and fetched in the usual way whenever a branch is desired during an application of this method. The auxiliary use of the history table does not provide any protection of entry addresses, which is the problem to be solved by the invention.

In a first embodiment of the method according to the invention, there is provided an area of reserved memory cells containing correlated data. The correlated data are separate from the program code and therefore not provided within the same individual instruction. If a program instruction is intended to perform a conditional jump to a certain entry address of a subroutine or function, there will be no previous consultation of branch history lists, contrary to

the method of Liu. The memory cells immediately before or after the entry address contain the address at which the correlated data are stored within the reserved memory area. If the reserved memory area, for example, contains all the addresses of the allowed entry points, the correlated data may again be the address to which the intended jump is directed. In those cases in which the data that are found at the address stored before or after the intended entry address are identical to the entry address, the correlation is affirmed and the entry address is recognized as allowed, and the jump is performed. In all other cases there are no correlated data, and therefore the jump instruction is interrupted or cancelled. In this simple example the same entry address is stored at the correlated data address that is stored before or after the entry address. The invention of the instant application encompasses different and more complicated choices of correlations between the data to be checked. The important feature of the method of the instant application is that immediately before or after the entry address, the address of correlated data is stored. Therefore, the correlation between the entry address and the provided correlated data stored in the reserved part of the memory can immediately be checked to verify whether the entry address is allowed. The necessary address of the correlated data is stored in the vicinity of the entry address and need

not be retrieved from an address table at a separate location.

There are alternative methods taught in the instant application that do not use reserved memory areas. They use other methods to compute and check data that are used for a correlation with the intended entry address. The specification of the instant application describes a special embodiment in which all the allowed entry addresses are preceded by the stored value of a special chosen function of the following fixed number of bytes. If this value is checked to be correct, which results in a positive correlation of the following bytes with the stored value, the entry address is verified as an allowed one. Such correlations can also occur accidentally. At any position within the program code there might be a stored value that is equal to the value of the chosen function calculated on the basis of the subsequent bytes. Therefore, the number of bytes that are used to compute the value of an appropriately chosen function is chosen greater than the maximum number of bytes that form one program instruction. In order to remove unwanted coincidences, no operation code is inserted by the compiler into the program instruction code to change the result of a computation of the function value at the respective location. A stored value and the value of the

function of the subsequent bytes will then only be equal at the allowed entry addresses. It is possible to arrange things accordingly, if the function value is computed from a number of bytes exceeding the maximum number of bytes exceeding the maximum number of bytes in the program instructions. Obviously, here the data are not provided within the same individual instruction.

Alternatively or additionally, the entry address can be protected by a kind of mark at the entry point that is formed by inserting a special byte sequence that cannot occur within the regular program codes (SNOP). If this is the sole methodology used to protect the entry addresses, the data to be correlated are the entry address and the special byte sequence in all the cases. Therefore, the correlation process in this embodiment may be reduced to the mere check that the special byte sequence is stored adjacent the entry address.

The method according to the instant application does not prognosticate program instructions by use of an updated table of previous branches, but is only concerned with the verification of the allowable entry addresses. The correlated data and their addresses have already been stored before the program is run or compiled. There is no

subsequent storage of correlated data or addresses of correlated data after the calculation of the correlation in the course of the execution of the program instructions, and there is no need to update and search tables of allowable entry addresses. The inventive improvements are completely different from the disclosure of Liu, and make use of principally different methodology. The method according to the instant application could not have been found by any modifications of the method described in Liu.

Turning now to the claim language of the instant application. The last paragraph of claim 1 recites:

> storing, in a memory cell, an address of a correlated data item one of directly before and directly after the permissible entry address. (emphasis added).

It is respectfully stated, and contrary to the Examiner's position, that this step is not taught in Liu. In the instant application the correlation of data takes place with data that are stored directly before or directly after the entry address, which is clearly different from the use of a separate programmed table of program information.

Claim 5 and 9 are directed to embodiments concerned with correlation of program data in non-reserved memory areas and a specific non-regular byte sequence or specific no-operation

code. This is not taught in Liu because specific code of this kind is not mentioned and would probably be useless in the branch history table methodology taught in Liu. More specifically, Liu is not believed to teach verifying that the entry address is permissible, rather Liu tries to predict where the next address will be. Put another way, Liu tries to predict the next step, whereas the invention of the instant application verifies that the chosen step has not been manipulated.

It is accordingly believed to be clear that none of the references, whether taken alone or in any combination, either show or suggest the features of claims 1, 5 or 9. Claims 1, 5 and 9 are, therefore, believed to be patentable over the art. The dependent claims are believed to be patentable as well because they all are ultimately dependent on claims 1, 5 or 9.

In view of the foregoing, reconsideration and allowance of claims 1-5 and 8-10 are solicited.

If an extension of time is required, petition for extension is herewith made. Any extension fee associated therewith should be charged to the Deposit Account of Lerner and Greenberg, P.A., No. 12-1099.

Please charge any other fees that might be due with respect to Sections 1.16 and 1.17 to the Deposit Account of Lerner and Greenberg, P.A., No. 12-1099.

Respectfully submitted,

For Applicants

REL:cgm

July 20, 2004

Lerner and Greenberg, P.A.
P.O. Box 2480
Hollywood, Florida 33022-2480
Tel.: (954) 925-1100
Fax: (954) 925-1101